

WHAT IS CLAIMED IS:

- 1                   1.     A method usable in an active router to route received packets,  
2     said method comprising the steps of:  
3                    associating threads with received packets for processing the received  
4     packets; and  
5                    while processing a previously received packet,  
6                    checking for the arrival of an interrupt;  
7                    creating a thread for associating said interrupt;  
8                    determining whether the thread associated with the interrupt has a  
9     priority that is higher than the priority of a thread associated with said previously  
10    received packet;  
11                   if the thread associated with the interrupt has a higher priority than  
12    said previously received packet, saving the thread associated with the previously  
13    received packet in a Shared Arena storage area;  
14                   if the thread associated with the interrupt does not have a higher  
15    priority than said previously received packet, queuing the thread associated with  
16    the interrupt.
- 1                   2.     The method according to claim 1, wherein the interrupt is an  
2     event indicating the arrival of a packet or expiration of a timer.
- 1                   3.     The method according to claim 1, wherein a thread is  
2     associated with each received packet or a group of received packets.
- 1                   4.     The method according to claim 1 further comprising a step of  
2     processing said thread associated with the interrupt, wherein the Shared Arena is  
3     accessible during said step of processing a previously received packet, said step of  
4     determining whether the thread associated with the interrupt has a priority that is  
5     higher than the priority of a thread associated with said previously received packet,  
6     and said step of processing said thread associated with the interrupt.

1           5.     The method according to claim 1, wherein the thread  
2 associated with the previously received packet saved in the Shared Arena is  
3 preempted by the interrupt having a higher priority, and the processing of the  
4 received packet is suspended in the Shared Arena.

1           6.     The method according to claim 1 further comprising the step  
2 of processing the interrupt.

1           7.     The method according to claim 6, wherein during said step of  
2 processing of the interrupt, further interrupts of lower or equal priority are  
3 disabled.

1           8.     The method according to claim 6, wherein when said step of  
2 processing of the interrupt has ended, the method further comprises the steps of:  
3                 determining whether there is a pending interrupt or thread having a  
4 higher priority than the thread saved in the Shared Arena;

5                 if there is a pending interrupt or thread having a higher priority than  
6 the thread saved in the Shared Arena, processing the next interrupt or thread; and,

7                 if there is no next interrupt or thread having a higher priority,  
8 resuming the processing of the thread associated with the previously received  
9 packet saved in the Shared Arena.

1           9.     The method according to claim 8, wherein prior to resuming  
2 the processing of the thread, the method further comprises the step of setting an  
3 identifier of a currently running thread.

1           10.    The method according to claim 1, wherein said step of  
2 associating threads with received packets further comprises the step of enqueueing  
3 said threads to a nonblocking priority run queue accessible for parallel access.

1           11.    The method according to claim 10, wherein said run queue  
2 includes an age value and a pointer that are updated with an operation to either add  
3 or remove a thread from said run queue, and said age value is used only to ensure  
4 one parallel operation at a time and a pointer indicating either an adding or  
5 removing of a thread.

1                   12.    The method according to claim 10, wherein said run queue is  
2   an array of nonblocking Last-In-First-Out ("LIFO") or First-In-First-Out ("FIFO")  
3   data structures.

1                   13.    A system usable in an active router to route received packets  
2   comprising of:

3                   a packet priority level process scheduling said threads and  
4   processing and routing the packets according to their priority;

5                   an interrupt priority handling process for handling an interrupt and  
6   associating threads with received packets and scheduling said packets during a  
7   processing of a previously received packet associated to a thread; and,

8                   a Shared Arena for storing the thread associated with the previously  
9   received packet before the processing of the interrupt;

10                  wherein said Shared Arena is a communication mechanism between  
11   said packet priority level packet process and said interrupt priority handling  
12   process.

1                   14.    The system as defined in claim 13 further comprising a  
2   nonblocking priority run queue accessible for parallel access.

1                   15.    The system as defined in claim 13, wherein the thread saved  
2   in the Shared Arena is suspended until the processing of the interrupt has ended.

1                   16.    The system as defined in claim 13, wherein the thread saved  
2   in the Shared Arena may be resumed when returning to packet priority level  
3   processing.

1                   17.    The system as defined in claim 16 further comprising a  
2   plurality of processors, and the interrupt is processed on one processor and the  
3   resumed thread is processed on another processor.

1                   18.    A router for routing received packets, said router comprising  
2   a set of instructions to:

3                   associating threads with received packets for processing the received  
4   packets; and

5                   while processing a previously received packet,  
6                   checking for the arrival of an interrupt;  
7                   creating a thread for associating said interrupt;  
8                   determining whether the thread associated with the interrupt has a  
9 priority that is higher than the priority of a thread associated with said previously  
10 received packet;  
11                  if the thread associated with the interrupt has a higher priority than  
12 said previously received packet, saving the thread associated with the previously  
13 received packet in a Shared Arena storage area;  
14                  if the thread associated with the interrupt does not have a higher  
15 priority than said previously received packet, queuing the thread associated with  
16 the interrupt.